

8º Workshop

ROBOTS DE EXTERIORES

ROBOCITY2030-II



Editores:

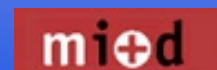
Roemi E. Fernández

Héctor Montes

Centro de Automática y Robótica
CSIC-UPM



Diciembre 2010



CAPÍTULO 13

DETECCIÓN E IDENTIFICACIÓN DE OBJETOS MÓVILES EN SISTEMAS MULTI-ROBOT CON INFORMACIÓN 3D

G. RODRIGUEZ-CANOSA, J. DEL CERRO y A. BARRIENTOS

Centro de Automática y Robótica UPM-CSIC, gonzalo.rcanosa@upm.es
j.delcerro@upm.es, Antonio.barrientos@upm.es

La detección y el seguimiento de objetos dinámicos es un aspecto crítico en los robots diseñados para realizar labores de vigilancia y seguridad. La utilización de sensores láser 3D permite la detección de objetos de cualquier forma y tamaño. En este contexto se han desarrollado dos métodos de detección de objetos, uno para las situaciones en las que el robot permanece inmóvil y otro para robots en movimiento. Asimismo se han desarrollado dos algoritmos que permiten identificar estos objetos y compartir la información entre diferentes robots para su integración en un sistema multi-robot. Los algoritmos han sido validados en un demostrador virtual desarrollado dentro del proyecto europeo NM-RS.

1 Introducción

La vigilancia de grandes infraestructuras es un aspecto muy importante de la seguridad para cualquier estado o empresa. Por esa razón se está impulsando cada vez más el desarrollo de soluciones robóticas para este tipo de misiones, tanto dentro del ámbito civil como el militar. Centrales de energía, centros de distribución de alimentos, infraestructuras de transportes e instalaciones de almacenamiento o producción de materiales peligrosos (químicos, biológicos o nucleares) se encuentran dentro de la lista de posibles emplazamientos de este tipo de desarrollos tecnológicos.

Los desarrollos comerciales actuales de soluciones robotizadas para sistemas de vigilancia y seguridad en el ámbito civil han experimentado grandes mejoras en los últimos años (MoviRobotics, 2010). Sin embargo,

todavía presentan deficiencias en la detección e identificación de objetos móviles dentro del perímetro de seguridad. La mayor parte de los desarrollos utilizan video para detectar diferencia entre fotogramas (Fernández-Caballero, 2003) o técnicas de reconocimiento de contornos activos (Tabb, 2002). También hay desarrollos basados en láseres de barrido horizontal (Fod, 2002), en los que se utilizan mapas de celdillas (Chen, 2008) o filtros de partículas (Montmerlo, 2002). El uso de sensores láser de barrido horizontal y vertical no está todavía muy extendido, por lo que se han desarrollado métodos de detección para láser 3D. El uso de esta tecnología permite la detección de objetos en todo el entorno, sin estar limitados a los planos de acción de los sensores de barrido horizontal; aunque introduce la dificultad añadida de tener que distinguir los objetos del suelo y el cielo.

Para la correcta implementación de un sistema robótico de seguridad es necesaria la integración de todos los robots, junto con sensores de diferente tipo e incluso algún supervisor humano, en un sistema híbrido que pueda funcionar de forma autónoma o semi-autónoma combinando información de diferentes fuentes. En este sentido (Stroupe, 2001) presenta un método para fusionar distintas medidas de un mismo objeto o (Mazo, 2004) que usan un equipo de dos robots para seguir a un sólo objeto móvil. El objetivo del trabajo es diseñar un algoritmo de fusión que permita compartir información sobre objetos dinámicos a diferentes tipos de robots; terrestres o aéreos, fijos o móviles, mejorando así las opciones disponibles actualmente.

2 Detección de objetos dinámicos

La detección de objetos dinámicos difiere si el sensor se encuentra en una plataforma móvil o en una plataforma estática. Si bien algunos desarrollos (Bobruk, 2002) utilizan el mismo método para los dos casos suponiendo movimientos pequeños; al pretender utilizar en este caso información 3D es necesario desarrollar dos métodos distintos.

2.1 Detección desde una plataforma estática.

En este caso se ha utilizado la diferencia directa de las matrices de distancia producidas por el sensor láser. Cada matriz de distancias producida por el láser es sometida al mismo proceso de diferencia, seguido por la aplicación de un filtro para detectar los puntos en los que la distancia ha cambiado más de un determinado valor umbral (Ecuación (1)), tal y como se observa en la Fig. 1.

$$Dif = \begin{cases} 1, si |(D_{ij} - D'_{ij})| > Dif_{lim} \\ 0, si |(D_{ij} - D'_{ij})| < Dif_{lim} \end{cases} \quad (1)$$

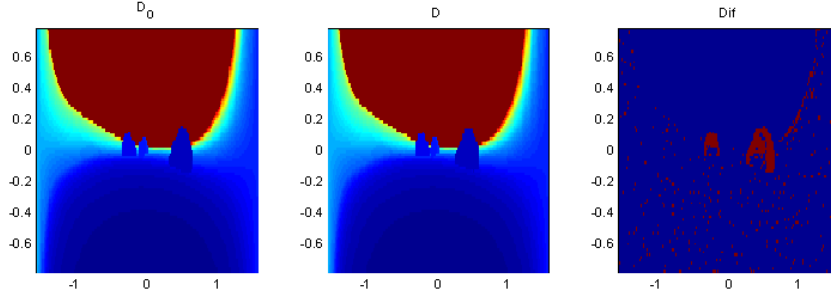


Fig. 1. Resultado de la diferencia de matrices tras la aplicación del filtro

Una vez se ha realizado la diferencia se agrupan los puntos mediante una conectividad a 8, asignándoles una etiqueta común e identificándolos como un mismo objeto. A continuación se eliminan todos los objetos de menos de 4 puntos, ya que se supone que se han producido por errores de medida o por pequeños cambios en el entorno.

Todos los puntos incluidos dentro de un mismo objeto son aquellos en los que se ha producido una diferencia en la distancia medida por el láser mayor que un determinado valor. Sin embargo, con este criterio, no todos los puntos diferencia pertenecen al objeto. En efecto, a medida que un objeto se desplaza, los puntos pertenecientes a su estela pueden quedar marcados como puntos del objeto pese a que la medida proporcionada por el láser corresponde en realidad con un punto del entorno. Para eliminar estos puntos se mantiene a lo largo de todas las iteraciones una matriz de referencia dinámica que se actualiza en función de la media \bar{D}_{ij} y de la varianza V_{ij} de las tres última matrices de distancia según la Ecuación (2).

$$\begin{aligned} D_{ij}^{ref} &= \bar{D}_{ij} \text{ y } V_{ij}^{ref} = V_{ij} \quad , \quad si \ V_{ij} < V_{ij}^{ref} \\ D_{ij}^{ref} &= \bar{D}_{ij} \text{ y } V_{ij}^{ref} = V_{ij} \quad , \quad si \ (V_{ij} < 4 \cdot V_{ij}^{ref} \wedge \bar{D}_{ij} > D_{ij}^{ref}) \end{aligned} \quad (2)$$

Con los puntos de la matriz que pertenecen realmente al objeto es necesario calcular características reales del mismo, como su tamaño y su posi-

ción. Para ello se calcula la posición real de los puntos con respecto al robot. Esto es posible ya que se conocen los ángulos del láser asociados a cada punto de la matriz, así como la distancia a cada punto de impacto. Asumiendo que se conoce la posición del robot con respecto al sistema de coordenadas global, es posible realizar una serie de cambios de sistemas de referencia para obtener la posición de todos los puntos de cada objeto.

La posición de cada objeto se obtiene realizando la media entre las celdas de la matriz pertenecientes a cada objeto, determinando así la celda central. La posición del punto de colisión de esa celda es al que se toma como posición real del objeto.

Para determinar el tamaño se calculan el máximo y el mínimo en cada coordenada entre todos los puntos que forman parte de cada objeto. Esta evaluación se realiza con las coordenadas reales, no con las celdas de la matriz. Tan sólo dos de esas medidas son significativas debido a las características intrínsecas a un sensor láser.

Al final de este proceso se obtiene una lista con todos los objetos detectados por el láser, con información sobre su posición y tamaño.

2.2 Detección desde plataforma móvil

La diferencia directa de matrices de distancia no es válida en este caso puesto que estas matrices varían mucho con pequeños desplazamientos del robot. Por lo tanto se ha optado por comparar reconstrucciones del terreno obtenidas a partir de medidas del láser.

Una reconstrucción es un mapa de celdas de varias capas en las que cada celda tiene asociado una serie de valores numéricos calculados a partir de los puntos de colisión del láser cuya proyección en el plano coincida dentro de esa celda. En este caso se han utilizado mapas con capas de confianza, altura máxima y gradiente en X y en Y. Un ejemplo para la capa de alturas de una reconstrucción se muestra en la Fig. 2.

Es necesario utilizar varias capas para eliminar falsos positivos producidos, por ejemplo, por objetos que entren progresivamente en el rango de detección del robot de forma que la altura de determinadas celdas iría aumentando sin la presencia de un objeto dinámico. Las capas consideradas en este análisis son: capa de confianzas, capa de gradientes y capa de alturas.

La capa de confianzas se utiliza para tener una medida de la fiabilidad de los datos almacenados en cada celda. Puesto que la reconstrucción se realiza a partir de las medidas obtenidas con un láser de barrido vertical y horizontal con incrementos de paso muy pequeños, se obtiene un gran número de puntos de colisión. Se probaron dos métodos de cálculo de la

confianza, uno basado en el número de puntos de impacto y otro basado en la dispersión de los puntos de impacto dentro de cada celda. Se eligió el segundo procedimiento dado que resultó más eficaz en las pruebas realizadas.

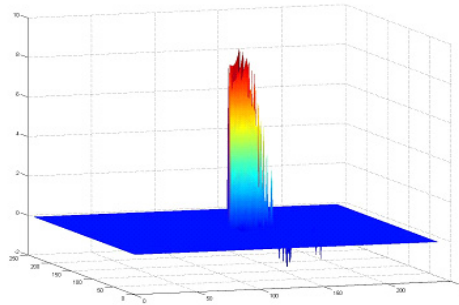


Fig. 2. Capa de alturas de una reconstrucción

La capa de gradientes tiene como objeto almacenar la estimación de las pendientes del terreno para cada celda según las orientaciones principales. El gradiente es calculado con las alturas de los puntos de colisión en una misma celda. Se utiliza para eliminar las detecciones de paredes y otros elementos estáticos a medida que el robot se acerca a ellos.

La capa más importante es la capa de alturas puesto que con ella se determina la existencia de movimiento en una celda determinada. En primer lugar se utilizó la altura media como indicador; pasando posteriormente a usar la altura máxima de la celda debido a que se obtuvieron mejores resultados.

Teniendo en cuenta esta evaluación de las distintas capas consideradas, es posible proceder a la comparación de reconstrucciones sucesivas para la obtención de mapas de movimiento. Si se comparan dos reconstrucciones sucesivas $(R_k \text{ y } R_{k-1})$ se obtiene un mapa de movimiento M_k , siguiendo las siguientes reglas:

- La confianza de cada celda debe ser mayor que un valor mínimo marcado previamente.
- La diferencia entre los gradientes X e Y deberá ser mayor que cero y menor que un valor que dependerá de la misión.
- La diferencia entre las alturas de los dos mapas consecutivos deberá ser mayor que un valor determinado, teniendo en cuenta que no se detectarán objetos de menos altura que esa tolerancia.

Experimentalmente se observó que disminuía el número de falsos positivos si se comparaban dos mapas de movimiento sucesivos, obteniéndose un mapa de movimiento más fiable. Llamando M_k al mapa de movimiento más reciente (obtenido a partir de (R_k, yR_{k-1})) y M_{k-1} al anterior (obtenido a partir de (R_{k-1}, yR_{k-2})), el proceso que se sigue para la obtención del mapa de movimiento final Mov es el siguiente:

- Si hay movimiento en el mapa de movimiento actual.
 $M_k(i, j) = 1$.
- Si hay movimiento en el entorno de la celda del mapa anterior $M_{k-1}(u, v) = 1; u = [i, i+1, i-1], v = [j, j+1, j-1]$
- Se marca movimiento en el mapa de movimiento final.
 $Mov(i, j) = 1$

Al inicio del movimiento del robot para comenzar el proceso de detección, se establece el centro del mapa coincidiendo con la posición del robot. Sin embargo, a medida que el robot se mueve hay que tener en cuenta que el alcance del robot quede dentro del mapa elegido. En el caso de que el rango de detección del robot quedase fuera del mapa sería necesario mover el centro del mismo, tal y como se observa en la Fig. 3. Por lo tanto al realizar todo el proceso de detección de objetos móviles hay que tener en cuenta que las tres reconstrucciones (R_k, R_{k-1}, yR_{k-2}) tengan el mismo centro. El proceso de detección no podría realizarse en las tres iteraciones siguientes a un cambio de centro de mapa.

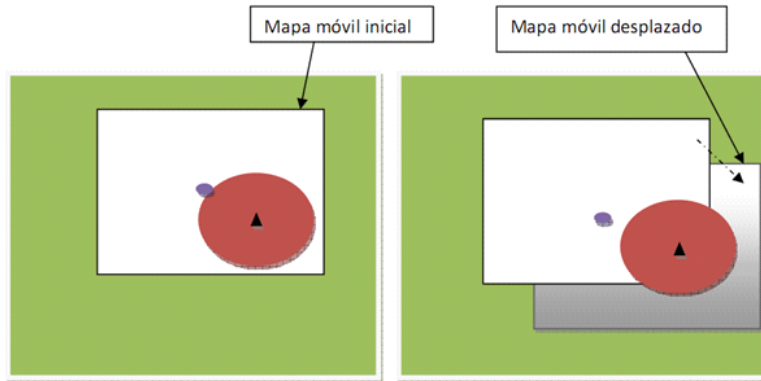


Fig. 3. Cambio de centro del mapa al desplazarse el robot debido a que el alcance del robot queda fuera del mapa original

Una vez obtenido el mapa de movimiento efectivo se agrupan las celdas siguiendo una conectividad a 8, obteniéndose así los objetos en movimiento. Se marca como posición de cada objeto el centro de la celda central de cada grupo.

Según el proceso descrito no se obtienen medidas de tamaño. Ello es así porque, si bien se podría realizar una media de los puntos de colisión de cada celda no existe una forma sencilla de separar los puntos que pertenecen al robot y los puntos que pertenecen al entorno.

3 Identificación de objetos. Fusión de listas

En una aplicación de seguridad en la que hay que vigilar una gran extensión de terreno es necesario que el sistema sea capaz de identificar correctamente a los objetos y de seguir sus trayectorias siempre que estén dentro del rango de detección. Este proceso se ha denominado identificación de objetos y consiste principalmente en fusionar la información de dos listas diferentes.

La identificación de objetos ocurre en diferentes partes del sistema simultáneamente, pero se pueden señalar dos situaciones tipo:

- Identificación de objetos **en un sólo robot**. Cada robot del sistema mantiene una lista local que va actualizando con cada lista de objetos proveniente de su propio módulo de detección. Este proceso se realiza cada vez que el robot obtiene una nueva medida del láser.
- Identificación de objetos **entre robots**. Los robots del sistema se comunican entre sí y con la estación central, enviándose las listas de objetos. Este proceso se puede dar tanto de forma centralizada (los robots sólo se comunican con la estación central), como de forma descentralizada (los robots se comunican entre sí y con la estación central).

La principal diferencia entre estas dos fusiones se refiere al tiempo transcurrido entre las medidas respectivas. En la identificación de objetos dentro de un mismo robot se conoce el tiempo transcurrido entre la media anterior y la actual y se puede garantizar que todos los valores de la lista entrante son más recientes que los almacenados en el robot. Cuando la fusión ocurre con listas provenientes de distintos robots no se puede garantizar que la información de la lista entrante sea más reciente que la almacenada en la lista local, ni es posible conocer de antemano el tiempo transcurrido entre las medidas.

Para solucionar los problemas específicos que se presentan en cada uno de los casos se han desarrollado dos algoritmos con la misma estructura común pero con algunas diferencias; según el caso considerado.

3.1 Identificación de objetos mediante filtro de kalman de estado variable.

Las plataformas de seguridad robóticas suelen integrar un gran número de sensores (MoviRobotics, 2010) para facilitar la tarea de vigilancia. En este caso sólo se incluye la detección de los objetos mediante un sistema láser, pero es lógico pensar que, una vez detectados, se dispondrá de más información sobre los objetos además de su posición y tamaño. Características como el color o la temperatura se pueden obtener fácilmente con los sensores disponibles actualmente y, tras la calibración de los mismos, se asigne a cada pixel un valor con técnicas de fusión de sensores (Kim, 2008).

El algoritmo diseñado empareja la lista proveniente del módulo de detección con la lista local almacenada en el robot, siguiendo el esquema que se presenta en la Fig. 4. Se crea un filtro de Kalman de estado variable en el que se utiliza un vector de estado con 7 variables por objeto considerado, como se muestra en la Ecuación (3).

$$\begin{aligned} x_i &= [x, y, size, R_{RGB}, G_{RGB}, B_{RGB}, Temp] \\ x_k &= [x_1, x_2, \dots, x_n] \end{aligned} \quad (3)$$

En cuanto a las matrices del modelo del filtro de kalman se utiliza un sistema lineal que supone un movimiento a velocidad constante mientras el resto de las variables permanecen constantes.

Para las covarianzas y las varianzas en el ruido del modelo y la medida (P_{ini} , R_{ini} y Q_{ini}); se utilizan valores obtenidos de las especificaciones de los sensores utilizados en cada caso.

La entrada del filtro sería la velocidad que se calcula con los dos estados anteriores, teniendo en cuenta que entre dos estados consecutivos un mismo objeto no mantiene la misma posición en el vector de estado.

Los pasos que sigue el algoritmo son los siguientes:

- Predicción del estado siguiente mediante el filtro de Kalman.
- Cálculo de todas las posibles parejas entre la nueva medida (los objetos en la lista de entrada) y el la predicción del estado actual (el estado predicho de los objetos de la lista local).
- Selección de las parejas con menor distancia de Mahalanobis.

- División de todos los vectores y matrices del filtro en función de si el objeto a que corresponde cada fila y columna ha sido emparejado o no.
- Iteración del filtro con los objetos emparejados.
- Recuperación del resto de variables e inclusión de los nuevos objetos.

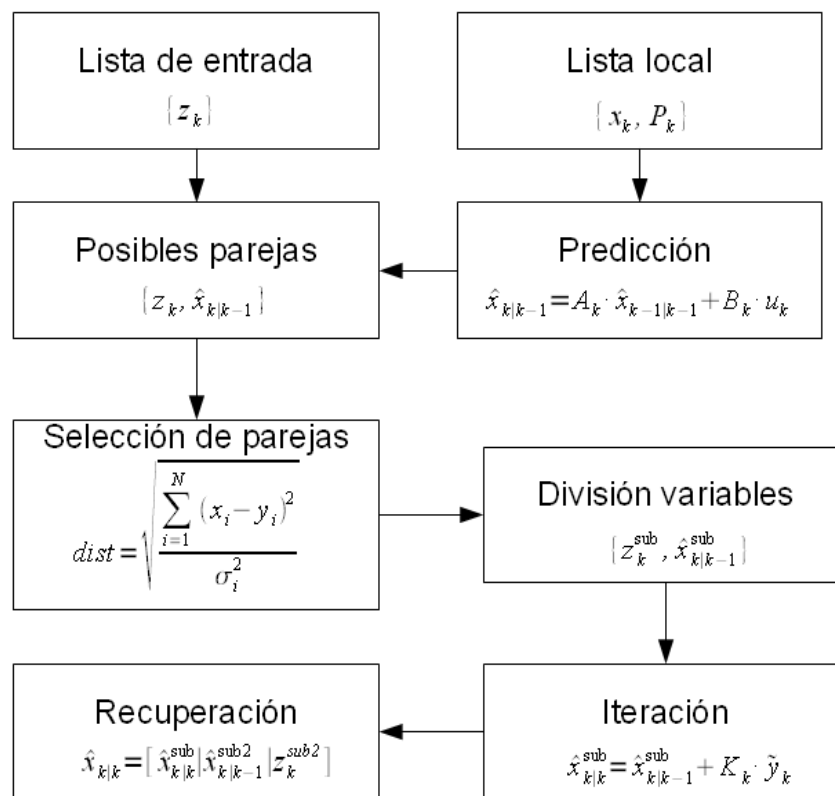


Fig. 4. Diagrama del algoritmo de identificación

3.2 Algoritmo de fusión de listas

La estructura de este algoritmo es similar al algoritmo anterior aunque en este caso no se utiliza un filtro de Kalman. Este procedimiento se utilizará tanto cuando un robot le envíe su lista de objetos dinámicos a otro robot cómo cuando se la envíe a la estación central, que en este caso actúa como un robot más del sistema.

La principal dificultad del emparejamiento de objetos entre dos listas provenientes de distintos robots es que los tiempos de detección de cada objeto pueden ser muy diferentes. Esta posibilidad tiene que ser tenida en cuenta para el cálculo de la distancia entre los objetos que integran todas las posibles parejas.

- Se determina el número de objetos de cada lista; n_{loc} para la lista local y n_{ent} para la lista de entrada. Se calcularán $n_{loc} \times n_{ent}$ parejas distintas.
- Para cada pareja se comparan los tiempos de detección de los dos objetos y se extrapola la posición del objeto cuyo tiempo de detección sea menor; es decir, cuya medida sea menos reciente. Este proceso se lleva a cabo para todas las parejas, siempre y cuando no hayan transcurrido más de 30 segundos entre las medidas, tal y cómo se muestra en la Ecuación (4)

$$\text{Si } |t_{det}^{loc} - t_{det}^{ent}| < 15$$

$$\begin{aligned} t_{det}^{loc} > t_{det}^{ent} &\rightarrow x_{pred}^{ent} = x_{ent} + v_{ent} (t_{det}^{loc} - t_{det}^{ent}) \\ t_{det}^{loc} < t_{det}^{ent} &\rightarrow x_{pred}^{loc} = x_{loc} + v_{loc} (t_{det}^{ent} - t_{det}^{loc}) \end{aligned} \quad (4)$$

- Se calcula la distancia entre los objetos para todas las parejas y se ordenan las parejas en función de la distancia. Se seleccionan tantas parejas como objetos contenga la lista con un menor número de elementos.
- Se actualiza la información de la lista local, siempre y cuando la información de la lista de entrada sea más reciente.
- Se incluyen en la lista local los objetos de la lista de entrada que no fueron emparejados.
- Se eliminan de la lista local aquellos objetos que lleven mucho tiempo sin ser detectados.

Una descripción global del algoritmo se muestra en la Fig. 5.

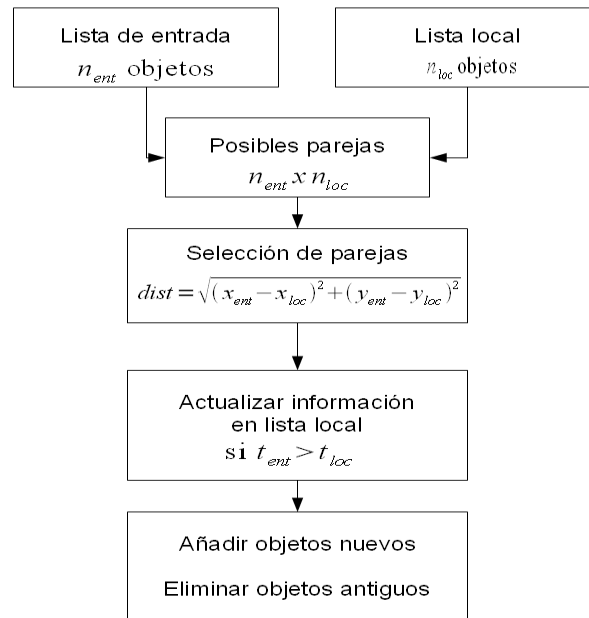


Fig. 5. Esquema del algoritmo de fusión de listas

4 Simulaciones

Los algoritmos mencionados en los apartados anteriores se han validado implementándolos dentro de un simulador desarrollado en el marco del proyecto europeo NM-RS (*Networked Multi-Robots Systems*). Este proyecto ha desarrollado un demostrador virtual para comprobar las capacidades de sistemas multirobot en misiones de vigilancia y rescate en entornos exteriores tanto urbanos como no estructurados.

Se ha realizado un gran número de pruebas de las que a continuación se presenta una selección para ejemplificar el funcionamiento del algoritmo.

4.1 Detección e identificación mediante el algoritmo de Kalman desde una plataforma estática

En este caso se utiliza un robot estático y tres robots en movimiento que hacen las veces de objetos dinámicos como se muestra en la Fig. 6, donde se indican las trayectorias de cada objeto.

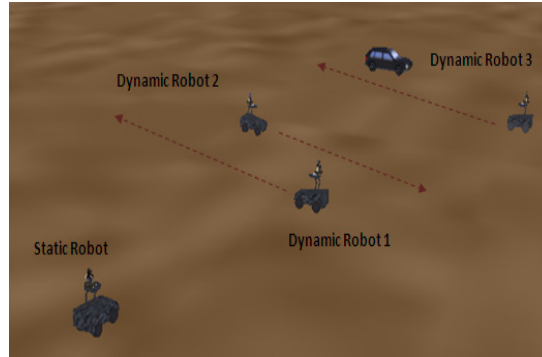


Fig. 6. Disposición inicial de los elementos involucrados en la prueba

En la Fig. 7 se muestran las trayectorias reales de los objetos y las trayectorias de los objetos detectados por el robot en una vista en planta.

Se observa que el algoritmo detecta tres objetos móviles y que los identifica correctamente a lo largo de toda su trayectoria. Incluso cuando se producen oclusiones (un objeto tapa la trayectoria del otro) el algoritmo es capaz de reconocerlos de forma efectiva. Hay que señalar que debido a las características matemáticas del filtro de Kalman es necesario un periodo de inicialización en el que los puntos tienen un error mayor.

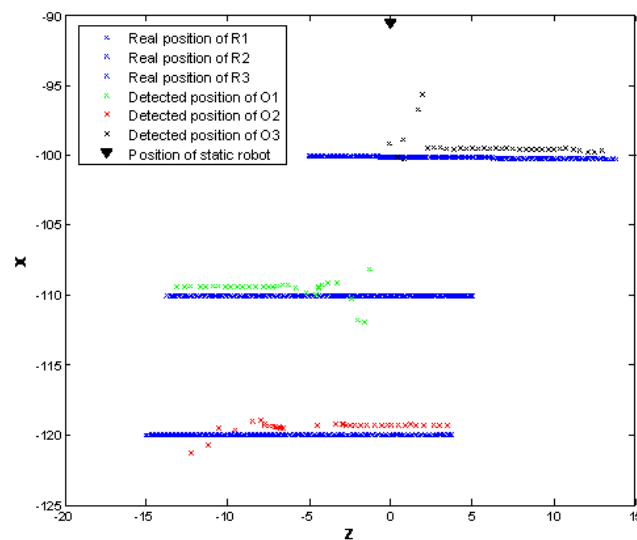


Fig. 7. Trayectorias en el plano XZ de los objetos y posiciones detectadas por el robot estático

4.2 Detección e identificación desde una plataforma en movimiento

En este caso se han usado dos robots en movimiento, uno de ellos actuará de detector y el otro de objeto móvil. Ambos llevarán trayectorias paralelas, aunque el objeto móvil irá describiendo curvas. Las trayectorias del robot móvil (en línea recta) y del objeto a detectar (línea ondulada), así como los puntos del mapa en los que el robot ha detectado movimiento se muestran en la Fig. 8.

Se observa cómo el robot ha detectado correctamente el contorno de todo el objeto móvil a pesar de que llevaba un movimiento no rectilíneo. En este caso se ha utilizado el algoritmo de fusión de listas como algoritmo de identificación y se ha comprobado que funciona correctamente, aunque no es capaz de incluir la información proveniente del resto de sensores.

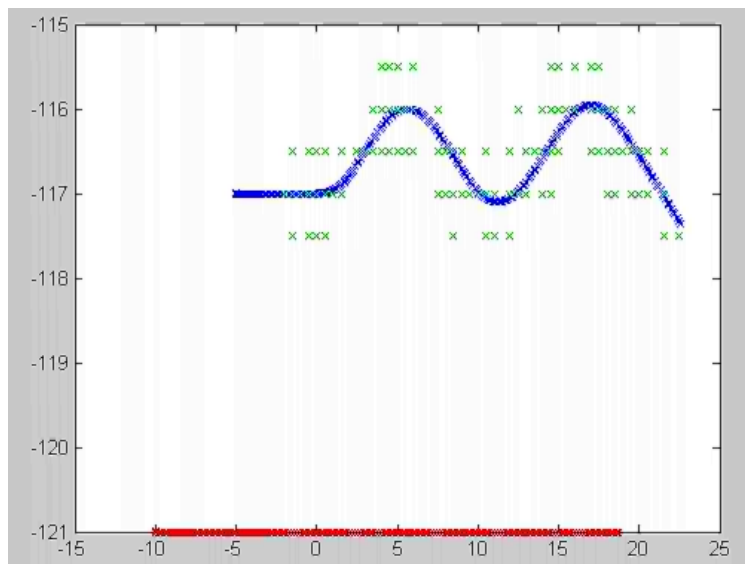


Fig. 8. Trayectorias del robot y el objeto; y puntos en los que el robot ha detectado movimiento

5 Conclusiones

En este trabajo se han desarrollado técnicas de análisis de las matrices de distancia producidas por un láser de barrido horizontal y vertical para la obtención de listas de objetos dinámicos. Se ha estudiado la diferencia en los métodos de análisis según si el sensor está inmóvil o en movimiento y se ha desarrollado un algoritmo de detección para cada caso.

También se ha estudiado la utilización de módulos de detección en aplicaciones de seguridad concluyéndose que es necesario poder identificar correctamente a los objetos para seguir sus trayectorias. Para ello se han desarrollado dos algoritmos de identificación. El primero de ellos identifica los objetos dinámicos detectados con una lista local al robot, combinando información proveniente de distintos sensores mediante un filtro de Kalman. El segundo método combina listas de objetos provenientes de distintos robots y logra identificar objetos a pesar de haber sido detectados en instantes muy diferentes.

Como trabajos futuros hay que señalar que, si bien se han probado los algoritmos de forma exhaustiva en un demostrador virtual muy completo, es necesario implementarlos en un sistema físico para ver su desempeño en tiempo real. Así mismo hay que estudiar la utilización de sensores Flash Lidar para reducir los tiempos de medida.

Agradecimientos

Este proyecto se ha realizado dentro del Grupo de Robótica y Cibernética de la Universidad Politécnica de Madrid. Se ha integrado dentro del proyecto europeo NM-RS / B-0004-ESM2_ERG de la Agencia de Defensa Europea. También pertenece a las actividades creadas en el marco de la red Robocity 2030-II de la Comunidad de Madrid.

Referencias

Ashley, W., Stroupe, M., Martin, C. and Balch, T. 2001. Distributed Sensor Fusion for Object Position Estimation by Multi-Robot Systems. Proceedings of the 2001 IEEE International Conference on Robotics & Automation. Seoul, Korea, pp. 21-26.

Bobruk, J. and Austin, D. 2004. Laser Motion Detection and Hypothesis Tracking from a Mobile Platform. Australasian Conference on Robotics and Automation.

Chen, B., Cai, Z., Xiao, Z. 2008. Real-time Detection of Dynamic Obstacle Using Laser Radar. 9th IEEE International Conference for Young Computer Scientists.

Fernandez-Caballero, A., Mira, J., Fernandez, M. A., Delgado, A. E. 2003. On motion detection through a multi-layer neural network architecture. Neural Networks, 16: 205-222.

Fod, A., Howard, A. and Matari, M. J. 2002. Laser-based people tracking. IEEE International Conference on Robotics and Automation, Washington D. C., pp. 3024-3029.

Kim, Y. *et al.* 2008. Design and calibration of a multi-view TOF sensor fusion system. In CVPR Workshops, pp. 1-7.

Mazo, M., Speranzon, A., Johansson, K. H. 2004. Multi-Robot tracking of a Moving Object Using Directional Sensors. Proceedings of the International Conference on Robotics and Automation, pp. 1103-1108.

Montemerlo, M., Thrun, S. and Whittaker, W. 2002. Conditional particle filters for simultaneous mobile robot localization and people-tracking. IEEE International Conference on Robotics and Automation, Washington, D. C., pp.695-701.

MoviRobotics. 2010. mSecurit, robot de vigilancia. Available at: <http://www.movirobotics.com/SPmsecurit.php>.

Tabb, K., Davey, N., Adams, R. and George, S. 2002. The recognition and analysis of animated objects using neural networks and active contour models. In Neurocomputing, Special edition on Engineering Applications on Neural Networks, vol. 43, pp. 145-172.

